



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED

DEC 03 2003

Applicant(s): Larry D. Barto, Steven C. Nettles, Yiwei Li

Assignee: Advanced Micro Devices, Inc.

Title: Starvation Avoidance Lost Start Agent (SALSA)

Technology Center 2100

Serial No.: 09/825,225 Filing Date: April 3, 2001

Examiner: Elliot L. Frank Group Art Unit: 2125

Docket No.: TT3418 Customer No.: 33438

Austin, Texas  
November 24, 2003

Mail Stop AF  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF UNDER 37 CFR § 1.191**

Dear Sir:

Applicant submits this Appeal Brief pursuant to the Notice of Appeal filed in this case on November 24, 2003.

Accompanying this Appeal Brief is a Petition to Restart a Previously Set Period for Reply and Conditional Petition for Extension of Time and a Notice of Appeal from the Decision of the Examiner. The Commissioner is hereby authorized to deduct any amounts required for this appeal brief and to credit any amounts overpaid to Deposit Account No. 502264. This paper is submitted in duplicate.

**I. REAL PARTY IN INTEREST**

The real party in interest is the assignee, Advanced Micro Devices, Inc. as named in the caption above.

## **II. RELATED APPEALS AND INTERFERENCES**

Based on information and belief, there are no appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals in the pending appeal.

## **III. STATUS OF CLAIMS**

Claims 1 – 26 are pending in the application. Claims 1 – 26 stand rejected. Claims 1 – 26 are appealed. Appendix “A” contains the full set of pending claims.

## **IV. STATUS OF AMENDMENTS**

All Amendments have been entered.

## **V. SUMMARY OF THE INVENTION**

The present invention relates to a system 120 which monitors work in process (WIP) in a manufacturing facility. The system includes a software object 200m that identifies a bottleneck workstation and a software object 200m that calculates a WIP value representing the amount of work approaching the bottleneck workstation. The system further includes a software object 200m that determines whether the WIP value is projected to fall below a control limit during an evaluation period and a software object that recommends, if the WIP value is projected to fall below the control limit during the evaluation period, that a selected amount of additional work be released into the manufacturing line. In at least one embodiment, the system further includes a software object 200m that determine one or more product types for the selected amount of additional work.

## **VI. ISSUE**

Are claims 1 – 26 allowable over Weaver et al., U.S. Patent No. 5,446,671 (Weaver).

## VII. GROUPING OF THE CLAIMS

For the purposes of this appeal, claims 1 – 26 may be grouped together.

## VIII. ARGUMENTS

### Claims 1 – 26 are allowable over Weaver

The present invention, as set forth by independent claim 1, relates to an automated system that monitors work-in-process (“WIP”) in a manufacturing facility. The system includes a software object that determines when an evaluation cycle should be invoked, and a recommendation wakeup listener object that performs the evaluation cycle. The recommendation wakeup listener object further includes a software object that identifies a bottleneck workstation, a software object that calculates *a WIP value representing the amount of work approaching the bottleneck workstation*, a software object that determines *whether the WIP value is projected to fall below a control limit during an evaluation period*, and a software object that recommends, if the WIP value is projected to fall below the control limit during the evaluation period, that *a selected amount of additional work be released into a manufacturing line of the manufacturing facility*.

The present invention, as set forth by independent claim 5, relates to an automated system that controls work-in-process (“WIP”) in a manufacturing facility. The system includes a software object that determines when an evaluation cycle should be invoked, and a recommendation wakeup listener object that performs the evaluation cycle. The recommendation wakeup listener object further includes an object that identifies a plurality of bottleneck workstations, an object that calculates *a WIP value for each of the plurality of bottleneck workstations*, wherein *each of the WIP values represents the amount of work approaching the corresponding bottleneck workstation*, an object that determines, for each WIP value, *whether the WIP value is projected to fall below a control limit during an evaluation period*, and an object that recommends, if any of the WIP values are projected to fall below the control limit during the evaluation period, that *a selected amount of additional work be released into a manufacturing line of the manufacturing facility*.

The present invention, as set forth by independent claim 8, relates to a method of controlling work-in-process (“WIP”). The method includes providing a software object that determines when an evaluation cycle should be invoked, and providing a recommendation wakeup listener object that performs the evaluation cycle. The providing recommendation wakeup listener object further includes providing a software object that identifies a bottleneck workstation, providing a software object that calculates *a WIP value representing the amount of work approaching the bottleneck workstation*, providing a software object that determines *whether the WIP value is projected to fall below a control limit during an evaluation period*, and providing a software object that recommends, if the WIP value is projected to fall below the control limit during the evaluation period, that *a selected amount of additional work be released into a manufacturing line*.

The present invention, as set forth by independent claim 14, relates to a method of controlling work-in-process (“WIP”). The method includes determining when an evaluation cycle should be invoked, and performing the evaluation cycle. The performing the evaluation cycle further includes identifying a bottleneck workstation, calculating *a WIP value representing the amount of work approaching the bottleneck workstation*, determining *whether the WIP value is projected to fall below a control limit during an evaluation period*, and recommending, if the WIP value is projected to fall below the control limit during the evaluation period, that *a selected amount of additional work be released into a manufacturing line*.

The present invention, as set forth by independent claim 20, relates to a manufacturing facility which includes a bottleneck workstation, and an automated system that monitors work-in-process (“WIP”) wherein the automated system includes a software object that determines when an evaluation cycle should be invoked, and a recommendation wakeup listener object that performs the evaluation cycle. The recommendation wakeup listener object further includes a software object that identifies the bottleneck workstation, a software object that calculates *a WIP value representing the amount of work approaching the bottleneck workstation*, a software object that determines *whether the WIP value is projected to fall below a control limit during an evaluation period*, and a software object that recommends, if the WIP value is projected to fall below the control limit during the evaluation period, that *a selected amount of additional work be released into a manufacturing line*.

The present invention, as set forth by independent claim 24, relates to a manufacturing facility which includes a plurality of bottleneck workstations, a software object that determines when an evaluation cycle should be invoked, and a recommendation wakeup listener object that performs the evaluation cycle. The recommendation wakeup listener object further includes an object that identifies the plurality of bottleneck workstations, an object that calculates a *WIP value for each of the plurality of bottleneck workstations, wherein each of the WIP values represents the amount of work approaching the corresponding bottleneck workstation*, an object that determines, for each WIP value, *whether the WIP value is projected to fall below a control limit during an evaluation period*, and an object that recommends, if any of the WIP values are projected to fall below the control limit during the evaluation period, *that a selected amount of additional work be released into a manufacturing line of the manufacturing facility*.

Weaver discloses a look-ahead method for determining optimum production schedules for each production step based on factory-wide monitoring of in-process part queues at all potential production bottlenecks. Weaver relates to a method for dispatching, i.e., the method is directed to trying to determine which lot should be dispatched next. (See e.g., Weaver Abstract.) Additionally, Weaver sets forth that the program functions in a sleep mode that is then interrupt driven. (See e.g., Col. 4, line 64 – Col. 5, line 15).

More specifically, Weaver relates to a computerized manufacturing control system “aimed at optimizing utilization of universal equipment that feeds a production step containing one or more potential bottlenecks.” (Weaver, col.1, lines 6-11). The operation of the control system of Weaver is discussed referencing Figure 3 of Weaver. More specifically, Weaver sets forth:

Because of size, the flow chart has been split into four interconnected sub-charts corresponding to FIGS. 3A-3D. In this flow chart, decision or conditional program steps within the logic flow chart are identified by the letter "D" (e.g., D1, D2, D3, etc.), while assignment or action program steps are identified by the letter "A" (e.g., A1, A2, A3, etc.). Starting in the upper left-hand corner of the chart (FIG. 3A), *the program is normally in a "sleep" mode, and is interrupt driven*. At least two interrupts are defined for the preferred implementation of the program. *The first interrupt (program step A1) is generated whenever new lots of any product are transferred to the step P queue. The second interrupt (program step A2) is generated whenever a resource (e.g., a machine) completes the processing of a product lot*.

Still referring to FIG. 3, either of the two interrupts prompts the program to query in program step D1 *whether or not a resource is idle*. A YES determination at D1 prompts the program to save both the resource ID and its setup ID step A3, thereby invoking program step A4; a NO determination invokes a query at program step D1 as to whether or not new work has been added to the Step P Queue. A NO determination at D2 returns the program to the "sleep" mode; a YES determination at D2 invokes program step A4.

Still referring to FIG. 3, step A4 selects the product that matches the saved setup ID on the idle resource. Program step D3 then determines *whether or not there is at least one lot of product in the step P queue* that matches the product selected in step A4. If the answer to the step D3 query is YES, program step D9 is invoked; if the answer to the D3 query is NO, *an alternate product must be selected from those awaiting processing at the step P queue*. In order to select the highest priority product, the priority search value (PSV) is set to zero, and a first search loop is initiated beginning with step D4. Step D4 accesses determines whether or not any product is in the step P queue which has the current priority search value. For this information, the program accesses the Program Data File. If the answer to program step D4 is YES, the program, in step A9 selects the product lot having the current PSV. Then, in step D8, the program determines whether or not a threshold quantity (Q.sub.THR) Of the selected product is available for processing. If Q.sub.AVL (the lot quantity of the selected product in the step P queue) is greater than Q.sub.THR, the program invokes step D9; if Q.sub.AVL is less than Q.sub.THR, the program invokes step A6. If the answer to program step D4 is NO, the program also invokes step A6. In step A6, the priority search value is incremented by one, and step D5 is invoked. Step D5 determines whether or not the priority search value exceeds P.sub.MAX. For the sake of simplicity, the value of P.sub.MAX will be determined by the number of different products. For example, if there are four products, each will have a different priority, and P.sub.MAX is 4. If the priority search value does not exceed P.sub.MAX, the program returns to step D4; if the priority search value exceeds P.sub.MAX, then the program will return to the sleep mode until interrupted.

Still referring to FIG. 3, upon the invocation of step D9, the program determines whether Q.sub.MIN and Q.sub.MAX values have been established for the selected product. If not, the program in step A11, selects the next lot of the currently selected product in the step P queue, and lot processing is begun when program Step A14 is invoked. When lot processing is complete, step A15 is invoked. Step A15 decrements the existing Q.sub.AVL value in the Program Data File, writes the step P queue data record for the completed lot to the step P+1 queue data file, and clears the Step P Queue Data File record for the completed lot.

If on the other hand, it is determined at step D9 that Q.sub.MIN and Q.sub.MAX values have been established for the selected product, program step A10 gets the Q.sub.MIN, Q.sub.MAX, Q.sub.NOW, and Q flag values for the selected product. Program step D10 then determines whether Q.sub.NOW is greater than Q.sub.MAX. A YES determination will cause the program to invoke step A11, which sets the Q flag for the selected product in the Program Data File, following which step, the program jumps to step A6. A NO

determination at step D10 invokes step D11, which determines whether or not Q.sub.NOW is less than Q.sub.MIN. A YES determination at D11 will clear the Q flag for the selected product in step A13, and the program will then jump to step A12. A NO determination at D11 will invoke step D12, which determines whether or not the Q flag is set for the selected product. If, so, the program invokes step A6; if not the program invokes step A12. (Weaver, col. 4, line 58- col. 6, line 11, emphasis added.)

Thus, Weaver is directed to determining when a resource is idle and then determining what to do with the idle resource whereas the present invention is directed to trying to avoid idle resources in the first place (i.e., starvation avoidance). There is no discussion in Weaver of releasing new work into a manufacturing line if a WIP value is projected to fall below a particular level. This differentiation can be seen throughout the claims, such as, the element which is substantially present in each independent claim which sets forth that “a selected amount of additional work be released into a manufacturing line of the manufacturing facility” if “the WIP value is projected to fall below the control limit during the evaluation period.”

More specifically, Weaver, taken alone or in combination, does not teach or suggest an automated system that monitors work-in-process (“WIP”) in a manufacturing facility which includes a software object that determines when an evaluation cycle should be invoked, and a recommendation wakeup listener object that performs the evaluation cycle much less such a system that includes a software object that recommends, *if the WIP value is projected to fall below the control limit during the evaluation period, that a selected amount of additional work be released into the manufacturing line*, all as required by independent claim 1. Accordingly, claim 1 is allowable over Weaver. Claims 2 – 4 depend from claim 1 and are allowable for at least this reason.

Weaver, taken alone or in combination, does not teach or suggest an automated system that controls work-in-process (“WIP”) in a manufacturing facility which includes a software object that determines when an evaluation cycle should be invoked, and a recommendation wakeup listener object that performs the evaluation cycle, much less such a system that includes an object that recommends, *if any of the WIP values are projected to fall below the control limit during the evaluation period, that a selected amount of additional work be released into the manufacturing line*, all as required by independent claim 5. Accordingly, claim 5 is allowable over Weaver. Claims 6 and 7 depend from claim 5 and are allowable for at least this reason.

Weaver, taken alone or in combination, does not teach or suggest a method of controlling work-in-process (“WIP”) which includes providing a software object that determines when an evaluation cycle should be invoked, and providing a recommendation wakeup listener object that performs the evaluation cycle, much less such a method that includes providing a software object that recommends, *if the WIP value is projected to fall below the control limit during the evaluation period, that a selected amount of additional work be released into the manufacturing line*, all as required by independent claim 8. Accordingly, claim 8 is allowable over Weaver. Claims 9 – 13 depend from claim 8 and are allowable for at least this reason.

Weaver, taken alone or in combination, does not teach or suggest a method of controlling work-in-process (“WIP”) which includes determining when an evaluation cycle should be invoked, and performing the evaluation cycle, much less such a method that includes recommending, *if the WIP value is projected to fall below the control limit during the evaluation period, that a selected amount of additional work be released into the manufacturing line*, all as required by independent claim 14. Accordingly, claim 14 is allowable over Weaver. Claims 15 – 19 depend from claim 14 and are allowable for at least this reason.

Weaver, taken alone or in combination, does not teach or suggest a manufacturing facility which includes a bottleneck workstation, and an automated system that monitors work-in-process (“WIP”) wherein the automated system includes a software object that determines when an evaluation cycle should be invoked, and a recommendation wakeup listener object that performs the evaluation cycle, much less such a facility that includes a software object that recommends, *if the WIP value is projected to fall below the control limit during the evaluation period, that a selected amount of additional work be released into the manufacturing line*, all as required by independent claim 20. Accordingly, claim 20 is allowable over Weaver. Claims 21 – 23 depend from claim 20 and are allowable for at least this reason.

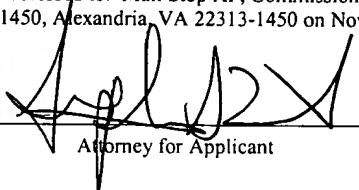
Weaver, taken alone or in combination, does not teach or suggest a manufacturing facility which includes a plurality of bottleneck workstations, a software object that determines when an evaluation cycle should be invoked, and a recommendation wakeup listener object that performs the evaluation cycle, much less such a system that includes an object that determines, for each WIP value, whether the WIP value is projected to fall below a control limit during an evaluation

period, and an object that recommends, *if any of the WIP values are projected to fall below the control limit during the evaluation period, that a selected amount of additional work be released into the manufacturing line*, all as required by independent claim 24. Accordingly, claim 24 is allowable over Weaver. Claims 25 and 26 depend from claim 24 and are allowable for at least this reason.

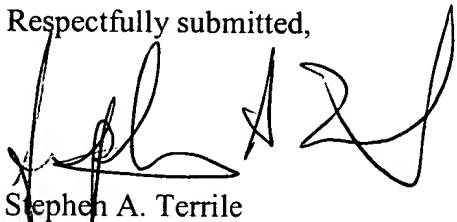
## IX. CONCLUSION

For the above reasons, Applicant respectfully submits that rejection of pending Claims 1-26 is unfounded. Accordingly, Applicant requests that the rejection of claims 1-26 be reversed.

This paper is submitted in duplicate.

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Mail Stop AF, Commissioner For Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on November 24, 2003.	
	11/24/03
Attorney for Applicant	Date of Signature

Respectfully submitted,

  
Stephen A. Terrile  
Attorney for Applicant  
Reg. No. 32,946

## APPENDIX "A"

1. (Previously Presented) An automated system that monitors work-in-process ("WIP") in a manufacturing facility, comprising:
  - a software object that determines when an evaluation cycle should be invoked; and
  - a recommendation wakeup listener object that performs the evaluation cycle, the recommendation wakeup listener object further including:
    - a software object that identifies a bottleneck workstation;
    - a software object that calculates a WIP value representing the amount of work approaching the bottleneck workstation;
    - a software object that determines whether the WIP value is projected to fall below a control limit during an evaluation period; and
    - a software object that recommends, if the WIP value is projected to fall below the control limit during the evaluation period, that a selected amount of additional work be released into a manufacturing line of the manufacturing facility.
2. (Original) The automated system recited in Claim 1, wherein the work approaching the bottleneck workstation comprises one or more product types.
3. (Original) The automated system recited in Claim 1, wherein the additional work comprises one or more product types.
4. (Original) The automated system recited in Claim 1 further comprises:
  - a software object that selects one or more product types for the selected amount of additional work.
5. (Previously Presented) An automated system that controls work-in-process ("WIP") in a manufacturing facility, comprising:
  - a software object that determines when an evaluation cycle should be invoked; and
  - a recommendation wakeup listener object that performs the evaluation cycle, the recommendation wakeup listener object further including:
    - an object that identifies a plurality of bottleneck workstations;

an object that calculates a WIP value for each of the plurality of bottleneck workstations, wherein each of the WIP values represents the amount of work approaching the corresponding bottleneck workstation ;

an object that determines, for each WIP value, whether the WIP value is projected to fall below a control limit during an evaluation period; and

an object that recommends, if any of the WIP values are projected to fall below the control limit during the evaluation period, that a selected amount of additional work be released into a manufacturing line of the manufacturing facility.

6. (Previously Presented) The automated system recited in Claim 5, wherein the additional work comprises one or more product types.

7. (Previously Presented) The automated system recited in Claim 5, wherein the work approaching the corresponding bottleneck workstation comprises one or more product types.

8. (Previously Presented) A method of controlling work-in-process (“WIP”), comprising:

providing a software object that determines when an evaluation cycle should be invoked;  
and

providing a recommendation wakeup listener object that performs the evaluation cycle,  
the providing recommendation wakeup listener object further includes:

providing a software object that identifies a bottleneck workstation;

providing a software object that calculates a WIP value representing the amount of work  
approaching the bottleneck workstation;

providing a software object that determines whether the WIP value is projected to fall  
below a control limit during an evaluation period; and

providing a software object that recommends, if the WIP value is projected to fall below  
the control limit during the evaluation period, that a selected amount of additional  
work be released into [the] a manufacturing line.

9. (Original) The method recited in Claim 8 further comprises:

providing a software object to select one or more product types for the selected amount

of additional work.

10. (Original) The method recited in Claim 8, wherein:  
providing a software object to identify a bottleneck workstation further comprises  
employing a software object to identify one or more of a plurality of bottleneck  
workstations.

11. (Original) The method recited in Claim 8, wherein :  
providing a software object to calculate a WIP value representing the amount of work  
approaching the bottleneck workstation further comprises employing a software  
object to calculate a WIP value for each of a plurality of bottleneck workstations,  
wherein each of the WIP values represents work approaching the corresponding  
bottleneck workstation.

12. (Original) The method recited in Claim 8 wherein:  
providing a software object to determine whether the WIP value is projected to fall below  
a control limit during an evaluation period further comprises employing a  
software object to determine whether any of a plurality of WIP values is projected  
to fall below the control limit during the evaluation period.

13. (Original) The method recited in Claim 8, wherein:  
providing a software object to recommend, if the WIP value is projected to fall below the  
control limit during the evaluation period, that a selected amount of additional  
work be selected for the bottleneck workstation further comprises employing a  
software object to recommend, if the WIP value associated with each of a  
plurality of bottleneck workstations is projected to fall below the control limit  
during the evaluation period, that a selected amount of additional work be  
released into the manufacturing line.

14. (Previously Presented) A method of controlling work-in-process (“WIP”),  
comprising:  
determining when an evaluation cycle should be invoked; and  
performing the evaluation cycle, the performing the evaluation cycle further including:

identifying a bottleneck workstation;  
calculating a WIP value representing the amount of work approaching the bottleneck workstation;  
determining whether the WIP value is projected to fall below a control limit during an evaluation period; and  
recommending, if the WIP value is projected to fall below the control limit during the evaluation period, that a selected amount of additional work be released into a manufacturing line.

15. (Previously Added) The method recited in Claim 14 further comprises:  
selecting one or more product types for the selected amount of additional work.
16. (Previously Added) The method recited in Claim 14, wherein:  
identifying a bottleneck workstation further comprises identifying one or more of a plurality of bottleneck workstations.
17. (Previously Added) The method recited in Claim 14, wherein:  
calculating a WIP value representing the amount of work approaching the bottleneck workstation further comprises calculating a WIP value for each of a plurality of bottleneck workstations, wherein each of the WIP values represents work approaching the corresponding bottleneck workstation.
18. (Previously Added) The method recited in Claim 14 wherein:  
determining whether the WIP value is projected to fall below a control limit during an evaluation period further comprises determining whether any of a plurality of WIP values is projected to fall below the control limit during the evaluation period.
19. (Previously Added) The method recited in Claim 14, wherein:  
recommending, if the WIP value is projected to fall below the control limit during the evaluation period, that a selected amount of additional work be selected for the bottleneck workstation further comprises recommending, if the WIP value associated with each of a plurality of bottleneck workstations is projected to fall

below the control limit during the evaluation period, that a selected amount of additional work be released into the manufacturing line.

20. (Previously Presented) A manufacturing facility, comprising:  
a bottleneck workstation; and  
an automated system that monitors work-in-process (“WIP”);  
wherein the automated system includes:  
a software object that determines when an evaluation cycle should be invoked; and  
a recommendation wakeup listener object that performs the evaluation cycle, the recommendation wakeup listener object further including:  
a software object that identifies the bottleneck workstation;  
a software object that calculates a WIP value representing the amount of work approaching the bottleneck workstation;  
a software object that determines whether the WIP value is projected to fall below a control limit during an evaluation period; and  
a software object that recommends, if the WIP value is projected to fall below the control limit during the evaluation period, that a selected amount of additional work be released into a manufacturing line of the manufacturing facility.

21. (Previously Added) The manufacturing facility recited in Claim 20, wherein the work approaching the bottleneck workstation comprises one or more product types.

22. (Previously Added) The manufacturing facility recited in Claim 20, wherein the additional work comprises one or more product types.

23. (Previously Added) The manufacturing facility recited in Claim 20, wherein the automated system further comprises:  
a software object that selects one or more product types for the selected amount of additional work.

24. (Previously Presented) A manufacturing facility, comprising:  
a plurality of bottleneck workstations;  
a software object that determines when an evaluation cycle should be invoked; and

a recommendation wakeup listener object that performs the evaluation cycle, the recommendation wakeup listener object further including:

an object that identifies the plurality of bottleneck workstations;

an object that calculates a WIP value for each of the plurality of bottleneck workstations, wherein each of the WIP values represents the amount of work approaching the corresponding bottleneck workstation;

an object that determines, for each WIP value, whether the WIP value is projected to fall below a control limit during an evaluation period; and

an object that recommends, if any of the WIP values are projected to fall below the control limit during the evaluation period, that a selected amount of additional work be released into [the] a manufacturing line of the manufacturing facility.

25. (Previously Added) The manufacturing facility recited in Claim 24, wherein the additional work comprises one or more product types.

26. (Previously Added) The manufacturing facility recited in Claim 24, wherein the work approaching the corresponding bottleneck workstation comprises one or more product types.